

Predictions and Interpretation for Random Effects Models

STAT 245

Our RE model for whale dive duration and sonar

```
rem4 <- glmmTMB(DurAvg ~ DepthAvg + TransClass +  
                SonarA + (1|TagID/TagDayPeriod),  
                data = zc_dives,  
                REML = FALSE)
```

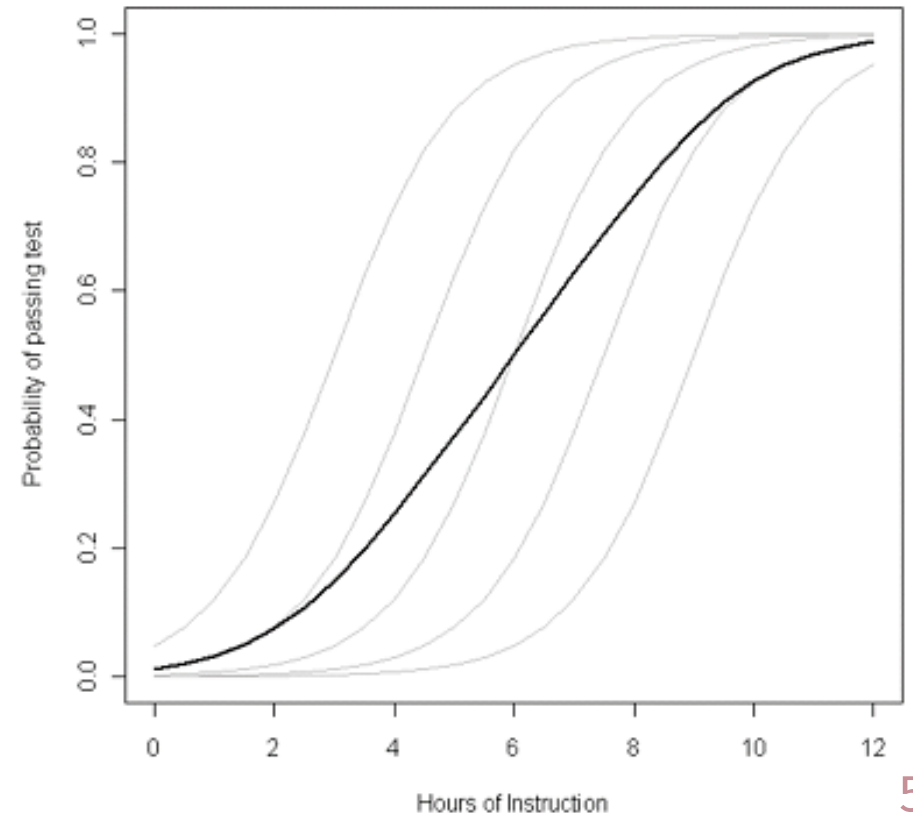
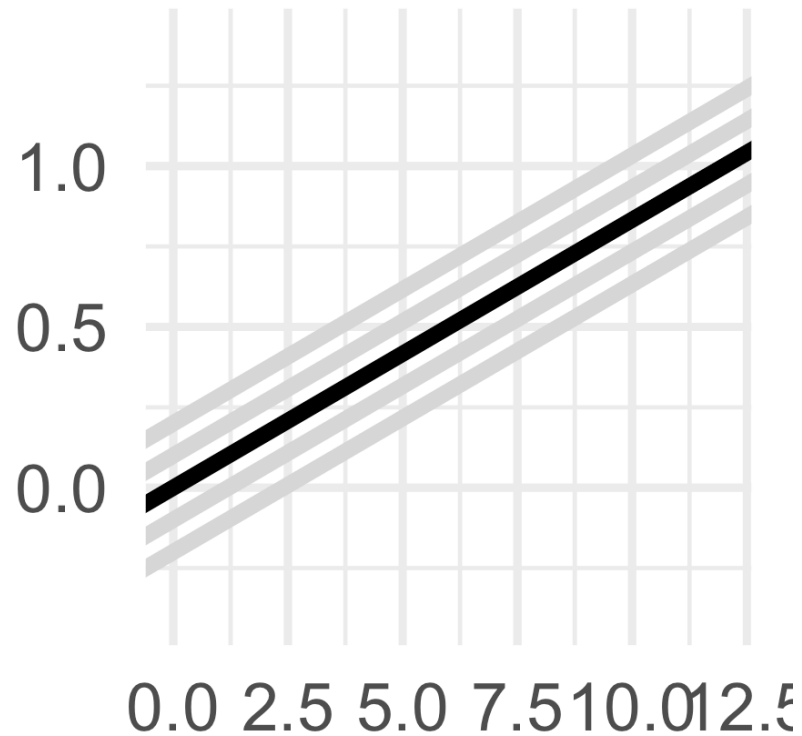
```
summary(rem4)
```

```
## Family: gaussian ( identity )
## Formula:
## DurAvg ~ DepthAvg + TransClass + SonarA + (1 | TagID/TagDayPeriod)
## Data: zc_dives
##
##           AIC          BIC   logLik deviance df.resid
##  36240.5   36301.1 -18111.3   36222.5      6174
##
## Random effects:
##
## Conditional model:
##   Groups              Name      Variance Std.Dev.
## TagDayPeriod:TagID (Intercept)  9.33     3.055
## TagID              (Intercept)  3.28     1.811
## Residual                                14.35     3.789
## Number of obs: 6183, groups: TagDayPeriod:TagID, 2143; TagID, 15
##
## Dispersion estimate for gaussian family (sigma^2): 14.4
##
## Conditional model:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   10.6143814  0.5522635   19.22 < 2e-16 ***
## DepthAvg       0.0399694  0.0005071   78.83 < 2e-16 ***
## TransClassDay  -0.1605083  0.2540850   -0.63 0.527576
## TransClassDusk -1.2486656  0.3384696   -3.69 0.000225 ***
## TransClassNight -2.3023853  0.2554556   -9.01 < 2e-16 ***
## SonarA1        2.8549212  0.6902209    4.14 3.53e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Typical (average) RE Group vs. Population Average

- Random effects models provide predictions for the *average or typical RE group*.
- **For a linear regression model (or any model with the identity link function, that is, no link function), the predicted values for the population average and typical-RE-group average are the same.**
- But with a link function in the mix, **it's different.**

Typical (average) RE Group vs. Population Average



Predictions with CIs for *Typical (average) RE Group*

Easily done, but *do not* include random effects variability

$$y = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k + \epsilon_{RE} + \epsilon_{resid},$$

$$\epsilon_{RE} \sim N(0, \sigma_{RE}); \epsilon_{resid} \sim N(0, \sigma_{resid})$$

Pred. plot w/ ggpredict()

```
library(ggeffects)
ggpredict(rem4,
          terms = 'SonarA',
          type = 'fixed') |>
plot() |>
gf_labs(y = 'Dive Duration (minutes)',
        title = '') |>
gf_lims(y = c(10, 35))
```

Pred. plot w/ `ggeffects :: ggpredict()`



BUT: What uncertainty is included here? Is ϵ_{RE} included?

Including ϵ_{RE}

Should you do this? Depends if want to include variation across RE groups (whale/hours)

```
ggpredict(rem4,  
          terms = 'SonarA',  
          type = 're') |>  
plot() |>  
gf_labs(y = 'Dive Duration (minutes)',  
        title = '') |>  
gf_lims(y = c(10, 35))
```

Including ϵ_{RE}

Should you do this? Depends if want to include variation across RE groups (whale/hours)



Parametric bootstrap to the rescue!

How can we get *population average* predictions?

- simulate data, based on our model, for (simulated) new RE groups. (Include uncertainty related to random effects, AND intercept/slope estimates)
- re-fit the model to the simulated data
- make predictions from the re-fitted-to-simulated-data model
- repeat LOTS of times to get a *distribution* of predicted values
- take a point-wise average over all those RE groups (and also use them to find a CI)
- result: population average predictions and confidence intervals.

Parametric Bootstrap

To obtain population-average predictions with CIs

We can do this with help from the function `bootMer()` from the `lme4` package.

Parametric Bootstrap

1. create function that makes predictions from our model.

```
predict_rem4 <- function(model){  
  new_dat <- data_grid(  
    model,  
    terms = 'SonarA',  
    condition = c(TagID = "14",  
                  TagDayPeriod = "2011-01-06.(18,20]")  
  )  
  return(predict(model,  
                newdata = new_dat,  
                type = "response",  
                allow.new.levels = TRUE,  
                re.form = NULL))  
}
```

Parametric Bootstrap

2. Simulate, make predictions for many new fake individuals

```
library(lme4)
# this will take a while
boot_rem4 <- bootMer(rem4,
                     FUN = predict_rem4,
                     nsim = 1000,
                     type = "parametric",
                     use.u = FALSE)
```

Simulation Results

```
glimpse(boot_rem4$t)
```

```
##   num [1:1000, 1:2] 22.6 13.2 19.3 19.9 15.9 ...
```

```
# show the first few rows of results
```

```
head(boot_rem4$t)
```

```
##           [,1]      [,2]  
## [1,] 22.58739 26.13058  
## [2,] 13.23180 15.08863  
## [3,] 19.31824 21.06538
```

Parametric Bootstrap

3. Compute CIs from simulated-individual predictions

```
new_data_pboot <- data_grid(rem4,  
                             terms = 'SonarA') |>  
  mutate(pred = apply(boot_rem4$t, 2, mean),  
         CIlow = apply(boot_rem4$t, 2,  
                       quantile, probs = 0.025),  
         CIhigh = apply(boot_rem4$t, 2,  
                        quantile, probs = 0.975)  
  )
```


Parametric Bootstrap

3. Compute CIs from simulated-individual predictions

```
glimpse(new_data_pboot)
```

```
## Rows: 2
## Columns: 8
## $ SonarA      <fct> 0, 1
## $ DepthAvg    <dbl> 293.2208, 293.2208
## $ TransClass  <fct> Night, Night
## $ TagDayPeriod <lgl> NA, NA
## $ TagID       <lgl> NA, NA
## $ pred        <dbl> 20.04807, 22.91534
## $ CIlow       <dbl> 14.90660, 17.68737
## $ CIhigh      <dbl> 25.15564, 28.25651
```

Parametric Bootstrap

4. Plot results

```
gf_point(pred ~ SonarA, data = new_data_pboot) |>  
  gf_labs(x='Sonar A Presence',  
          y='Dive Duration\n(minutes)') |>  
  gf_errorbar(CIlow + CIhigh ~ SonarA,  
              data = new_data_pboot, width = 0) |>  
  gf_lims(y = c(10, 35))
```

Parametric Bootstrap

4. Plot results (& compare w/ ave. RE group)

